Remarks

The present amendment responds to the Official Action dated November 8, 2005. A petition for a one month extension of the time to respond and authorization to charge Deposit Account No. 50-1058 the fee of $120 for this extension accompany this amendment. The Official Action rejected claims 1-20 under 35 U.S.C. 103(a) as unpatentable over Bishop et al. U.S. Patent No. 6,049,798 ("Bishop") in view of Barritz U.S. Patent No. 5,590,056 ("Barritz") in view of Farrell et al. U.S. Patent No. 5,247,675 ("Farrell"), in view of Yamagishi U.S. Patent No. 5,870,604 ("Yamagishi").

Claims 1, 2, 15 and 18 have been amended to be more clear and distinct. Claims 1-20 are presently pending.

The Art Rejections

The art rejections hinge on the application of Bishop, Barritz, Yamagishi and Farrell, taken in combination. As addressed in greater detail below, Bishop, Barritz, Yamagishi and Farrell do not support the Official Action's reading of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of Bishop, Barritz, Yamagishi and Farrell made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

The Official Action rejected claims 1-20 under 35 U.S.C. 103(a) as unpatentable over Bishop, Barritz, Yamagishi and Farrell. That action candidly admits that each item relied upon lacks one or more features of the claims as previously presented; however, it then concludes it would be obvious to pluck one piece of the claim from one reference and another from another

and so on. Such an approach smacks of an improper hindsight reconstruction of the claims. In light of the present amendments to claims 1, 2, 15 and 18, reconsideration and withdrawal of this rejection is respectfully requested.

Claim 1, as amended, claims that the operating software scheduling information capture software is operative to record a history of operating software schedule information "from within the operating system". See, page 1, lines 10 and 11, for example. To this end, the operating software scheduling information software "is an integral part of the operating system" and recording occurs" while the operating software is executing". See for example, page 3, lines 20-25. Similar amendments have been made in claim 15. Additionally, claims 2 and 18 have been amended to address aspects of the present invention described at page 12, lines 18-23, for example, which address a retail store bar code scanner embodiment in which "a specific bar code tag sequence" is scanned to select "the information to be recorded".

These combinations of features are not taught by Bishop as the Official Action correctly admits. Bishop teaches systems and techniques for monitoring of resources in a data processing system. Bishop monitors the overall level of resource usage, for example CPU and RAM usage, without concentrating on which individual tasks are using which resource. Bishop monitors peripheral device usage by receiving information from device drivers. The device drivers sometimes provide information about the process using the device, but the information provided by device drivers, and the other resource usage information provided by Bishop, does not include a history of operating software events with information relating to the history being organized and stored as operating software program scheduling information relating to interactions between

the operating system software and each of the programs and tasks managed by the operating system software, as is claimed by claim 1, as amended.

Bishop is directed toward the display of resource utilization in real time and contemplates the monitoring and displaying of resource utilization information during an operating session, with no indication that the collected is to be preserved for retrieval or use beyond the session during which it is collected. Bishop's designation of a viewing period allows for a user of the system of Bishop to specify a time window for which information is to be presented, but any past information collected for presentation within the time window is simply part of a continuous information stream, preserved so as to provide context for a user. This context allows a user to identify trends and changes and to see the effects changing conditions, including changes to the operation of the system caused by user actions and adjustments. Bishop does not indicate that the viewing period remains open past the termination of an operating session, and does not indicate that the collected information is retrievable past the termination of an operating session.

The storage of scheduling information claimed by claim 1 allows for retrieval and examination of information over a number of sessions. Such retrieval and examination allows for assembly and analysis of statistical information indicating system performance over a time period involving a number of sessions, and for comparison of system events and performance between different sessions. Analysis and comparisons made possible by preservation of scheduling information beyond the termination of a session allows for system refinements and improvements extending beyond those that can be performed during a single session. See, for example, the specification at p. 13, lines 17-23, which describes the retrieval of preserved

information and notes the usefulness of such information in the development of robust embedded software, such as in the embodiment of a retail bar code scanner as addressed at page 12, lines 8-23, for example, and claimed in presently amended claims 2 and 18.

Adding Barritz to Bishop does not cure Bishop's deficiencies as a reference with respect to claim 1, as amended. Claim 1 claims that a history of operating system of events organized and stored as scheduling information includes indications of relative priorities of programs and tasks, indications of transfers of control from lower priority to higher priority tasks and identification of tasks waiting for execution at the occurrence of each operating software event. Neither Bishop, Barritz nor a combination of Bishop and Barritz teaches or makes obvious these features. Bishop does not employ indications of relative priorities of programs and tasks. In order to track processor capacity that is not used by normally operating processes, Bishop begins a new process that is given a lower priority than any executing process on the system. Bishop notes the time employed by this lowest priority process in order to determine time that is not used by any of the normal processes on the system. The lowest priority process can only run when none of the normal processes are running, so the run time of the lowest priority process can be tracked and used to determine the normal idle time for the system, that is, the time when none of the normal processes on the system need to run. Bishop uses this lowest priority process to determine the overall system usage, and does not track the relative priorities of different processes.

Adding Barritz to Bishop does not achieve or make obvious the invention as claimed by claim 1. Barritz teaches the collection of frequency of usage information for various software

modules that may be hosted on a computer system. Barritz identifies a module when that module is invoked, identifying the particular software version represented by the module and matching module usage with licensing information in order to detect usage of unlicensed software. Barritz helps to identify underused and obsolete software modules in order to allow decisions to remove the underused and obsolete modules in order to save storage resources, and helps to identify unlicensed usage in order to assist in complying with licensing requirements. The information collected by Barritz does not include information relating to relative priorities of programs and tasks, transfers of control from lower priority to higher priority tasks and tasks waiting for execution at the occurrence of each operating software event. This information, collected by the present invention as claimed by claim 1, as amended, provides insight into the ongoing operation of a system and the allocation and management of resources to accomplish the various tasks performed by the system.

Adding Farrell to Bishop and Barritz does not teach or make obvious the invention as claimed by claim 1. Farrell teaches construction of threads having dispatch classes, and the storage and use of priority information in order to manage the assignment of priorities to threads and the allocation of processing resources among threads, but Farrell does not teach the relatively long term storage of scheduling information to permit review by a user, with the duration of storage being sufficient to allow data collected during an operating session to be retrieved and used after termination of the session. Storage of scheduling information to permit review by a user allows for insight by a user into the demands on and performance of a system and allows evaluation of the system to determine what adjustments, if any, need to be made, with the

evaluation being able to take into account the demands and performance prevailing in multiple operating sessions and comparisons between demands and performance across multiple sessions. Such insights, and the relatively long term storage of information that permits the sort of review leading to such insights, are not provided by or relevant to Farrell, which simply uses priority information in operation but does not store it in order that it may be delivered for review by a user.

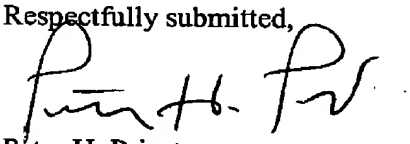
Adding Yamagishi to Bishop, Barritz and Farrell does not cure their deficiencies as references with respect to claim 1. Yamagishi distributes workloads among processors and uses data relating to the tasks awaiting execution and the number of tasks awaiting execution in order to manage workloads. Yamagishi does not collect data identifying the tasks awaiting execution at each particular software event for relatively long duration storage in order to permit review by a user, but simply uses the information in operation. Claim 1 claims recording scheduling information that includes identification of tasks awaiting execution at each software event so that the scheduling information may be presented to a user for review. Such collection and presentation of information allows for insight into the operation of a system and helps to determine what adjustments, if any, need to be made. The relatively long term storage for presentation to a user is not accomplished by or relevant to Yamagishi, which simply uses information relating to the tasks or numbers of task awaiting execution in workload balancing, but does not contemplate that the information will be reviewed in order to provide insight by a user into the operation of the system. Claim 1, as amended, therefore defines over the cited art and should be allowed.

A similar analysis applies to claim 15, and none of the relied upon art appears to address the claimed combinations of features of claims 2 and 18.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Peter H. Priest
Reg. No. 30,210
Priest and Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919)806-1600